

Super Cerveau



Super Cerveau

PAN Liuyan , CHABALIER Andy

Projet Logique et web sémantique ENSISA 2A IR

Repository git du projet: [https://git.chabalier.com/snowert/Super_Cerveau]

I) Objectifs:

- Réaliser une interface graphique pour le quizz
- Interroger et apprendre à utiliser DBpedia en SPARQL depuis l'API Apache Jena
- Réaliser un quizz automatique qui construit des questions par le biais de requêtes DBpedia
- Proposer des questions avec un support visuel (Reconnaitre une image ect.)

II) Fonctionnement:

Réalisé en Swing, l'interface est composée de deux fenêtres, la fenêtre principale donne le choix du nombre de joueur et des différents sujets, puis lance la fenêtre de jeu. Après le démarrage de la partie, la gestion des questions est effectuée par la classe PanelQuestion

Lorsque un joueur atteint 10 points (Défini dans Constantes) le jeu s'arrête et propose de continuer pour atteindre un meilleur score ou s'arrêter.

Lors du lancement du programme, on initialise l'affichage et on attend la saisie des joueurs et des questions pour laisser WindowsPlayer prendre le relais. WindowsPlayer initialise l'interface de jeu. lorsque l'utilisateur lance la partie, PanelQuestion utilise la QuestionFactory pour créer les questions et réponses et permettre leur affichage

La factory initialise la nouvelle question. Celle ci interroge DBpedia par le biais d'une requête et récupère les informations nécessaires.

Lors d'une partie multijoueur, chaque joueur répond à une question différente, s'affrontant sur le score ou la gloire.

III) Categories de Questions:

- Acteur
- Capitale
- Championnat de football
- Chanteur
- Dessin Animé
- Film
- Peinture
- Race de chien
- Roi et prédécesseur
- Mélange (chaque question est tirée aléatoirement parmi les catégories de questions)

IV) Requêtes:

```
select ?nom ?comment where {
    ?animee a <http://dbpedia.org/ontology/Cartoon>.
    ?animee <http://www.w3.org/2000/01/rdf-schema#label> ?nom.
    ?animee <http://www.w3.org/2000/01/rdf-schema#comment> ?comment
    FILTER (lang(?nom) = 'fr')
    FILTER (lang(?comment) = 'fr')
}
```

```
select ?nomPeinture ?nomAuteur ?imgPeinture ?imgAuteur where { ?painting a
<http://dbpedia.org/ontology/Painting>.
    ?painting <http://dbpedia.org/ontology/author> ?auteur.
    ?painting <http://dbpedia.org/ontology/thumbnail> ?imgPeinture.
    ?painting <http://www.w3.org/2000/01/rdf-schema#label> ?nomPeinture.
    ?auteur <http://www.w3.org/2000/01/rdf-schema#label> ?nomAuteur.
    ?auteur <http://dbpedia.org/ontology/thumbnail> ?imgAuteur.
    FILTER (lang(?nomPeinture) = 'fr').
    FILTER (lang(?nomAuteur) = 'fr').
}
```

```
select ?nom (str(?imgg) as ?img) where {
    <http://fr.dbpedia.org/resource/Liste_des_races_de_chiens>
<http://dbpedia.org/ontology/wikiPageWikiLink> ?chien.
    ?chien <http://www.w3.org/2000/01/rdf-schema#label> ?nom.
    ?chien <http://dbpedia.org/ontology/thumbnail> ?imgg.
    FILTER (lang(?nom)='fr').
}
```

```
select ?nom ?type where {
    ?films a <http://dbpedia.org/ontology/Film>.
    ?films <http://www.w3.org/2000/01/rdf-schema#label> ?nom.
    ?films <http://dbpedia.org/ontology/genre> ?genre.
    ?genre <http://www.w3.org/2000/01/rdf-schema#label> ?type.
    FILTER (lang(?nom) = 'fr')
    FILTER (lang(?type) = 'fr')
}
```

```
select ?nom (str(?imgg) as ?img) where {
    ?acteur a <http://dbpedia.org/ontology/Actor>.
    ?acteur <http://dbpedia.org/ontology/thumbnail> ?imgg.
    ?acteur <http://www.w3.org/2000/01/rdf-schema#label> ?nom.
    FILTER (lang(?nom) = 'fr').
}
```

```
select ?nom (str(?imgg) as ?img) where {
    ?acteur a <http://dbpedia.org/ontology/Singer>.
    ?acteur <http://dbpedia.org/ontology/thumbnail> ?imgg.
    ?acteur <http://www.w3.org/2000/01/rdf-schema#label> ?nom.
    FILTER (lang(?nom) = 'fr').
}
```